

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) A method of running a virtual machine monitor on computer hardware, the hardware including memory, the method comprising:  

~~waiting to commence~~ commencing virtualization of the memory ~~until at~~  
multiple times during runtime of an operating system.
2. (Original) The method of claim 1, wherein the virtualization includes constructing an Identity mapping of physical to machine memory; and commencing to use the virtual machine monitor at runtime to manage memory translation.
3. (Original) The method of claim 2, wherein the Identity mapping is constructed prior to runtime.
4. (Original) The method of claim 2, wherein the memory translation is initially performed according to the Identity mapping.
5. (Original) The method of claim 4, wherein the virtual machine monitor modifies the mapping after the physical memory has been virtualized.
6. (Currently amended) The method of claim 2, ~~wherein an operating system is running on the virtual machine monitor prior to virtualizing the memory; and~~ wherein the memory translation is managed by allowing the operating system to define virtual-to-physical mapping, and the virtual machine monitor to define physical-to machine mapping.

7. (Original) The method of claim 6, wherein the virtual machine monitor dynamically composes virtual-to-physical translations with physical-to-machine translations.

8. (Original) The method of claim 6, wherein the virtual machine monitor inspects the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

9. (Original) The method of claim 6, further comprising loading a translation lookaside buffer with virtual-to-machine translations.

10. (Previously presented) The method of claim 1, wherein only a portion of the memory is virtualized at runtime.

11. (Original) The method of claim 1, wherein the hardware includes a CPU that was virtualized prior to the virtualization of the memory.

12. (Original) The method of claim 1, further comprising performing runtime devirtualization of the virtualized memory.

13. (Currently amended) A method of running a virtual machine monitor on computer hardware and an operating system on the virtual machine monitor, the hardware including memory, the memory virtualized by the virtual machine monitor, the method comprising:

devirtualizing the memory at runtime of an operating system.

14. (Original) The method of claim 13, wherein a portion of the memory is devirtualized.

15. (Original) The method of claim 13, wherein when the operating system is booted, the virtual machine monitor exposes the booting operating system to

physical memory no larger than machine memory, where the physical memory does not span any memory holes.

16. (Previously presented) The method of claim 13, wherein the operating system defines virtual-to-physical translations prior to the runtime devirtualization; wherein the virtual machine monitor defines physical-to-machine translations prior to the runtime devirtualization; wherein the virtual machine monitor composes dynamically the virtual-to-physical translations with the physical-to-machine translations prior to the runtime devirtualization, wherein the runtime devirtualization includes having the virtual machine monitor cease to perform the dynamic composition of translations.

17. (Original) The method of claim 13, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using the operating system to manage address translation with respect to the devirtualized memory.

18. (Original) The method of claim 17, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

19. (Original) The method of claim 17, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

20. (Original) The method of claim 17, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

21. (Original) The method of claim 20, further comprising preventing the physical-to-machine mapping from being modified during the remapping, and temporarily preventing some or all write accesses to memory.

22. (Original) The method of claim 17, wherein the operating system and any application activity is paused while the remapping is performed.

23. (Original) The method of claim 17, further comprising maintaining a back map that contains for each page of machine memory a list of the pages of physical memory that map to it, and a list of free machine pages.

24. (Original) The method of claim 17, wherein the remapping is performed without a back map by maintaining a reference count for each machine page is kept, and freeing machine pages when their reference counts are zero.

25. (Original) The method of claim 17, wherein the remapping is performed without a back map by constructing a list of the physical pages mapping to a page of machine memory by searching the physical-to-machine mapping.

26. (Original) The method of claim 17, wherein managing the address translation includes having the virtual machine monitor cease to inspect the operating system's virtual-to-physical translations; and ceasing to maintain a page table of direct virtual-to-machine mappings.

27. (Original) The method of claim 17, wherein managing the address translation includes having the virtual machine monitor cease to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.

28. (Currently amended) A computer comprising memory including first and second portions, the first portion encoded with a virtual machine monitor that waits to commence virtualization of the second portion until multiple times during runtime of an operating system.

29. (Original) The computer of claim 28, wherein the virtualization includes constructing an identity mapping of physical to machine memory; and commencing to use the virtual machine monitor at runtime to manage memory translation.

30. (Original) The computer of claim 29, wherein the virtual machine monitor modifies the mapping after the physical memory has been virtualized.

31. (Original) The computer of claim 29, wherein an operating system is running on the virtual machine monitor prior to virtualizing the memory; and wherein the memory translation is managed by allowing the operating system to manage virtual-to-physical mapping, and allowing the virtual machine monitor to manage physical-to-machine mapping.

32. (Original) The computer of claim 31, wherein the virtual machine monitor dynamically composes virtual-to-physical translations with the physical-to-machine translations.

33. (Original) The computer of claim 31, wherein the virtual machine monitor inspects the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

34. (Original) The computer of claim 31, wherein a translation lookaside buffer is loaded with the virtual-to-machine translations.

35. (Original) The computer of claim 28, wherein only a portion of physical memory is virtualized at runtime.

36. (Currently amended) An article for a computer, the article comprising computer memory including a first portion encoded with a virtual machine monitor

that ~~waits to~~ commences virtualization of a second portion of the memory ~~until~~ multiple times during runtime of an operating system.

37. (Original) The article of claim 36, wherein the virtualization includes constructing an Identity mapping of physical to machine memory; and commencing to use the virtual machine monitor at runtime to manage memory translation.

38. (Original) The article of claim 37, wherein the virtual machine monitor can modify the mapping after the physical memory has been virtualized.

39. (Original) The article of claim 37, wherein the memory translation is managed by allowing an operating system to manage virtual-to-physical mapping, and allowing the virtual machine monitor to manage physical-to machine mapping.

40. (Original) The article of claim 39, wherein the virtual machine monitor can dynamically compose virtual-to-physical translations with the physical-to-machine translations.

41. (Original) The article of claim 39, wherein the virtual machine monitor can inspect the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

42. (Original) The article of claim 37, wherein the virtual machine monitor can load a translation lookaside buffer with virtual-to-machine translations.

43. (Original) The article of claim 36, wherein the virtual machine monitor can virtualize only a portion of physical memory at runtime.

44. (Currently amended) A computer comprising hardware including memory; and a virtual machine monitor for virtualizing the memory and devirtualizing the memory at runtime, wherein the virtual machine monitor commences virtualization of ~~virtualizes~~ the memory when multiple operating system instances are running and devirtualizes the memory when a single operating system instance is running.

45. (Original) The computer of claim 44, wherein a portion of the memory is devirtualized.

46. (Original) The computer of claim 44, wherein when an operating system is booted, the virtual machine monitor exposes the booting operating system to physical memory no larger than machine memory, where the physical memory does not span any memory holes.

47. (Previously presented) The computer of claim 44, wherein an operating system defines virtual-to-physical translations prior to the runtime devirtualization; wherein the virtual machine monitor defines physical-to-machine translations prior to the runtime devirtualization; wherein the virtual machine monitor composes dynamically the virtual-to-physical translations with the physical-to-machine translations prior to the runtime devirtualization; wherein the runtime devirtualization includes having the virtual machine monitor cease to perform the dynamic composition of translations.

48. (Original) The computer of claim 44, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using an operating system to manage address translation with respect to the devirtualized memory.

49. (Original) The computer of claim 48, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

50. (Original) The computer of claim 48, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

51. (Original) The computer of claim 48, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

52. (Original) The computer of claim 51, wherein the physical-to-machine mapping is prevented from being modified during the remapping, and some or all write accesses to memory are temporarily prevented.

53. (Original) The computer of claim 48, wherein the operating system and any application activity is paused while the remapping is performed.

54. (Original) The computer of claim 48, wherein managing the address translation includes having the virtual machine monitor cease to inspect the operating system's virtual-to-physical translations; and wherein maintenance of a page table of direct virtual-to-machine mappings is ceased.

55. (Original) The computer of claim 48, wherein managing the address translation includes having the virtual machine monitor cease to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.



56. (Currently amended) An article for a computer including hardware, the hardware including computer memory, the article comprising memory encoded with software for devirtualizing the computer memory at runtime of an operating system.

57. (Original) The article of claim 56, wherein the software causes a portion of the memory to be devirtualized.

58. (Original) The article of claim 56, wherein the software includes a virtual machine monitor; and wherein when an operating system is booted on the virtual machine monitor, the virtual machine monitor exposes the booting operating system to physical memory no larger than machine memory, where the physical memory does not span any memory holes.

59. (Original) The article of claim 56, wherein an operating system defines virtual-to-physical translations prior to the runtime devirtualization; wherein the software includes a virtual machine monitor for defining physical-to-machine translations prior to the runtime devirtualization, composing dynamically the virtual-to-physical translations with the physical-to-machine translations prior to the runtime devirtualization, and ceasing to perform the dynamic composition of translations during the runtime virtualization; and wherein after the runtime devirtualization is performed, memory translation is performed by directly using the virtual-to-physical mapping defined by the operating system.

60. (Original) The article of claim 56, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using an operating system to manage address translation with respect to the devirtualized memory.

61. (Original) The article of claim 60, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

62. (Original) The article of claim 60, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

63. (Original) The article of claim 60, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

64. (Original) The article of claim 63, wherein the physical-to-machine mapping is prevented from being modified during the remapping, and some or all write accesses to memory are temporarily prevented.

65. (Original) The article of claim 60, wherein the operating system and any application activity is paused while the remapping is performed.

66. (Original) The article of claim 60, wherein the software includes a virtual machine monitor that manages the address translation by ceasing to inspect the operating system's virtual-to-physical translations; and wherein maintenance of a page table of direct virtual-to-machine mappings is ceased.

67. (Original) The article of claim 60, wherein the software includes a virtual machine monitor for managing the address translation by ceasing to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.